Exam 98-362
TA: Windows Development Fundamentals

Exam Design

The Basics

This is a Microsoft Technology Associate (MTA) exam designed to assess candidates' fundamental knowledge of and skills with Windows® development. MTA is a new certification under the Microsoft Certification Program that validates the foundational knowledge needed to begin building a career in Microsoft technologies. It can also serve as a stepping stone to the Microsoft Certified Technology Specialist exams. Successful candidates for this exam will earn an MTA certification as well as access to benefits of the Microsoft Certification Program. The primary target audience for the MTA certification is students attending high schools and two-year colleges.

We are specifying an item pool of 75 unique multiple-choice items, which will be used on 1 form. Approximately 70 percent of the items should be written to the knowledge/comprehension level and about 30 percent to the application level. For more information about cognitive levels, refer to the Cognitive Domain in Bloom's Taxonomy.

Categories in the cognitive domain of Bloom's Taxonomy (Anderson & Krathwohl, 2001)

The following anatomy is required of each knowledge-level item in this exam:
- Question Statement
- Answer Choices (Preferably Multiple Choice, choose 1, with 3 distracters)
  - Ex: Which of the following is a valid ASP.NET variable name? A. _foo; B. &foo; C. foo#; D. foo 1

The following anatomy is required of each application-level item in this exam:
- Concise scenario, including any constraints/requirements necessary to make distracter answers 100% incorrect

- Goal Statement (You need to …)
- Question Statement (ex: What should you do?)
- Answer Choices (Preferably Multiple Choice, choose 1, with 3 distracters)

Target Audience

Candidates for this exam are seeking to prove Windows programming and application development knowledge and skills. Before taking this exam, candidates should have a solid foundational knowledge of the topics outlined in this preparation guide. It is recommended that candidates be familiar with the concepts of and have hands-on experience with the technologies described here by either by taking relevant training courses or by working with tutorials and samples available on MSDN® and in Microsoft® Visual Studio®. Candidates are expected to have some experience with a Microsoft .NET language such as C# or Microsoft Visual Basic® .NET.

Candidates for this exam are in the process of expanding their knowledge and skills in the following areas:

- various types of applications that run on Windows

- how to create graphical user interface (GUI) applications that run on Windows by using Windows Forms or Windows Presentation Foundation (WPF)

- how Windows Services are programmed and hosted on a computer that runs Windows

- how to access data from various sources for use in a Windows-based application

- how to successfully deploy a Windows application to target computers

Objective Domain

1. **Understanding Windows Programming Basics**

    1.1. Identify Windows application types.
    This objective may include but is not limited to: Windows Forms, Windows Presentation Foundation (WPF), Windows Services, and Win32® applications

    1.2. Implement user interface design.
    This objective may include but is not limited to: core user interface design principles for creating graphical-based applications

    1.3. Create Windows-based applications by using Visual Studio.
    This objective may include but is not limited to: project types, importance of the various aspects of a Windows Application project

2. Creating Windows Forms Applications

2.1. Create and handle events.
This objective may include but is not limited to: methods for creating events in an application; handling events raised in an application

2.2. Understand Windows Forms inheritance.
This objective may include but is not limited to: implementing forms inheritance in applications for visual inheritance

2.3. Understand how to create new controls and extend existing controls.
This objective may include but is not limited to: creating a new GUI control or inheriting functionality from an existing control

2.4. Validate and implement user input.
This objective may include but is not limited to: implementing the correct user input model based on application design and requirements; accepting keyboard and mouse input; validating user input through GUI controls such as text boxes and dialog controls

2.5. Debug a Windows-based application.
This objective may include but is not limited to: using breakpoints and debugging techniques to identify issues in code; debugging a Windows Services application

3. Creating Windows Services Applications

3.1. Create a Windows Services application.
This objective may include but is not limited to: inheriting the ServiceBase class; writing code in the Main method; overriding the OnStart and OnStop procedures

3.2. Install a Windows Services application.
This objective may include but is not limited to: creating installers for Windows Services; installing services on a target computer

4. Accessing Data in a Windows Forms Application

4.1. Understand data access methods for a Windows Application.
This objective may include but is not limited to: connecting to a database

4.2. Understand databound controls.
This objective may include but is not limited to: how data is bound to controls; how to display the data in the appropriate manner; forms and WPF binding; validating databound items

5. Deploying a Windows Application

5.1. Understand Windows application deployment methods.
This objective may include but is not limited to: different methods of deploying Windows applications; choosing the appropriate method for deployment; deploying an application by using ClickOnce

.

5.2. Create Windows setup and deployment projects.
This objective may include but is not limited to: creating setup projects for applications; specifying custom actions; creating special folders; security requirements; x64 deployment and program files location